

# Passagem de dados para um script em PHP

Um script em PHP ou página HTML pode transferir dados para outro script em PHP no servidor de diversas maneiras. Algumas destas maneiras são:

- **Por meio de formulários, usando o método GET ou POST;**
- **Através da URL do navegador, montando uma query string (método GET por padrão);**
- **Por meio de um link inserido em uma página web (método GET por padrão).**

# Dados através da URL do navegador

- Uma aplicação web pode receber dados do cliente por meio de uma query string, passada pela URL do navegador. A query string deve ser composta dos seguintes elementos:
  1. Especificação do domínio hospeda o script e que irá receber os dados;
  2. Uma barra (/) seguida do nome do script que irá processar os dados;
  3. Uma interrogação (?) sendo seguida por um identificador, um sinal de igual (=) e o valor do dado a ser transmitido;
  4. Se houver dados adicionais, acrescenta-se um e comercial (&), seguido de um novo identificador, um sinal de igual e mais um valor a ser enviado. Pode-se repetir este passo 4) indefinidamente.
- Exemplo:

**www.meuservidor.com.br/processaDados.php?**  
**idade=35&altura=1.82&nascimento=10/07/1990**

# Dados enviados por um link

- Uma página web ou um script em PHP também pode enviar informações para outro script em PHP por meio de um link. A metodologia é muito semelhante ao uso da query string, visto no exemplo anterior. Veja:

```
<body>
```

```
<!-- passando dados por links -->
```

```
<a href="
```

```
http://www.meuservidor.com.br/processaDa  
dos.php?idade=35&altura=1.82&nascimento=  
10/07/1990"> Enviando informações ao
```

```
servidor </a>
```

```
</body>
```

# Recebendo os dados no servidor

- No servidor, o PHP receberá os dados usando um vetor superglobal chamado `$_GET`. O script em PHP no servidor que iria receber os dados de um link poderia ter a seguinte configuração:

```
<?php
//recebendo os dados do navegador
$idade = $_GET["idade"];
$altura = $_GET["altura"];
$nasc = $_GET["nascimento"];
?>
```

- O processor é exatamente o mesmo se os dados estivessem sendo enviados do cliente para o servidor por meio de um formulário com o **método get**.

# Solicitando dados pela URL

- Sempre que pedimos ao navegador para que o mesmo exiba determinada página ou script em PHP armazenados no servidor, o método padrão é o GET.
- Exemplo: se o usuário digitar, na URL do navegador, **<http://hdam.pro.br/PRW/mostra-medias.php>**, o PHP no servidor irá receber esta requisição por meio do método **GET**.

# Enviando dados ao servidor por meio de formulários

- Métodos GET ou POST;
- Se a tag <form> contiver o atributo method especificado com o valor get, o formulário enviará os dados por esse método;
- Se a tag <form> contiver o atributo method especificado com o valor post, o formulário enviará os dados pelo método POST;
- Se a tag <form> não contiver o atributo method, o padrão utilizado pelo navegador é GET;
- Exemplo:

**A) <form action="script.php" method="get">**

Neste caso, o servidor receberá os dados pelo método GET

**B) <form action="script.php" method="post">**

Neste caso, o servidor receberá os dados pelo método POST

# Limitações do método GET

- Insegurança no envio das informações: os dados são transmitidos junto do nome do script que irá recebê-los no servidor, na forma de query string, e aparecem na URL do navegador;
- Limite na quantidade de informações a serem enviadas pelo método GET: **2048 caracteres** (2 KB);
- Porém, uma vantagem deste método é que as informações de um formulário podem ser guardadas no bookmark ou favoritos do navegador. Com POST isso não é possível. Veja mais em <http://marceloweb.info/principais-diferencas-entre-os-metodos-http-get-e-post/>

# Recebendo dados no servidor

- A maioria dos elementos de um formulário exige um atributo name, ao qual atribuímos determinado valor;
- No servidor, o script em PHP utilizará estes "names" para identificar o objeto que está enviando o dado e salvar a informação. Para isso, a linguagem PHP usa dois vetores: \$\_POST ou \$\_GET;
- O vetor a ser utilizado depende do método escolhido para envio das informações. Para o método GET, o PHP salva a informação assim:

**`$nome = $_GET["nome"] ;`** e para o método POST:

**`$idade = $_POST['idade'] ;`** //o PHP admite apóstrofos também. O índice "nome" pode ser substituído por uma variável do PHP.

Elemento	HTML	PHP
textarea	<code>&lt;textarea name="sugestao"&gt;&lt;/textarea&gt;</code>	<code>\$var = \$_POST["sugestao"];</code>
text	<code>&lt;input type="text" name="idade" /&gt;</code>	<code>\$var = \$_POST["idade"];</code>
password	<code>&lt;input type="password" name="senha" /&gt;</code>	<code>\$var = \$_POST["senha"];</code>
file	<code>&lt;input type="file" name="arquivo" /&gt;</code>	<code>\$var = \$_POST["arquivo"];</code>
radio	<code>&lt;input type="radio" name="logar" value="permanecer logado" /&gt;</code>	<code>\$var = \$_POST["logar"];</code>
checkbox	<code>&lt;input type="checkbox" name="linguagens[]" value="PHP" /&gt;</code>	<code>\$var = \$_POST["linguagens"];</code>
select único	<code>&lt;select name="férias"&gt;</code>	<code>\$var = \$_POST["ferias"];</code>
select múltiplo	<code>&lt;select name="praias[]" multiple="multiple"&gt;</code>	<code>\$var = \$_POST["praias"];</code>

**Se o formulário usar GET, basta trocar \$\_POST por \$\_GET**